

ACE, THE ATARI CASSETTE ENHANCER

USING ACE - SOME GENERAL POINTS

Just before you look at the commands in the three ACE options, here are some general notes which apply to the whole operation and use of ACE. Including them here saves repeating them several times later.

- * ACE recognises your commands by a very simple command code character: the '@' sign. Therefore every time you ask ACE to do something, you must begin your ACE command with '@'. In addition, the '@' must be at the extreme left of the line which the cursor is on (column 1 is the usual way of describing it).
- * The '@' has been chosen because it isn't otherwise likely to appear in column 1. If you'd like to change '@' to something else, you may and it's very easy to do. Simply poke memory location 2047 with the ATASCII value of the alternative character you want to use. The ATASCII values of every character are in your Atari Basic Reference Manual. As an example, suppose you want the command code character to be '!'. Your Atari Basic Reference Manual shows you that the ATASCII value for '!' is 33. After you've loaded your chosen ACE option you therefore type POKE 2047,33 and hit 'RETURN'. This will immediately change the ACE command code character from '@' to '!'.

You can't permanently change the command code character, so remember that it will revert to '@' the next time you load ACE.

- * In the next few pages, you'll see the various ACE commands. When you want ACE to do something, there's no need to enter the whole command. Simply position the cursor in column 1, type the command code character, and then the first three letters of the ACE command. Of course if you like typing you can type the whole command, but the first three letters are sufficient.

Now go on to the next pages which give you a detailed description of the commands in the three ACE options.

OPTION 'A' FACILITIES AND COMMANDS

Here are the commands and facilities when Option 'A' is loaded:

@CSAVE"Name" Saves a program in memory on to cassette complete with a name up to 16 characters long. Such programs can still be loaded with a normal CLOAD command.

@CLOAD"Name" Searches for a program with the specified name and loads it if found.

@VERIFY"Name" Searches for a program with the specified name and verifies it.

@CRUN"Name" Searches for a program with the specified name. Loads and runs it if found.

@CATALOG Searches an entire cassette. All named programs found are listed on the screen.

@VERIFY Verifies a program previously saved in the LIST"X" format.

@FLASH Causes the cursor to 'blink' on and off. The 'blink' rate can be controlled by POKE 2421,X where X is a number between 1 and 255. The most effective rates are obtained with only one bit set - 1,2,4,8,16,32, 64 and 128.

(Option 'A' details continued on the next page)

OPTION 'A' FACILITIES AND COMMANDS - continued

@FLASH"Z" Causes any character typed in reverse (by using the Atari logo key) to 'blink' in a variety of ways. The value of 'Z' in the command determines the effect:

- 0 the cursor only 'blinks'
- 1 reverse characters appear normal and 'blink'
- 2 reverse characters 'blink' between reverse and normal
- 3 reverse characters appear normal and 'blink' between normal and solid white
- 4 all characters, whether reverse or not, 'blink' between normal and upside down
- 64 reverse characters remain reversed and 'blink' between reverse and solid white

Reverse screen characters can be made to 'blink' while a BASIC program is running. FLASH must have previously been activated.

@OFF Disables all FLASH commands and returns everything to normal operation.

OPTION 'B' FACILITIES AND COMMANDS

When Option 'B' is chosen and loaded, all the facilities just described for Option 'A' are available, together with these additional facilities:

- #DUMP** Prints an on-screen list of all variables used in the program currently in memory. The list only includes those variables used when the DUMP command is given. The values will be those which the variables had at that time. If the variables have been previously saved with a name (see Option 'A'), they will be included in the variables list.
- #AUTO** Activates the auto line numbering function. If there is no program in memory, auto numbering starts from 100. If there is a program in memory, auto numbering will begin at the last program line number plus a specified increment. If no increment is specified, the default is 10.
- #AUTO s,i** Activates the auto line numbering function. Values should be entered for 's' and 'i'. Auto numbering will start at the value given to 's' and increment in steps of the value given to 'i'.
- #AUTO s** Activates the auto line numbering function. A value should be entered for 's'. Auto numbering will start at the value given to 's' and increment in steps of 10.
- If AUTO is activated and numbering should reach a line number which already exists (if, for example, a new section is being inserted in the middle of an existing program in memory), AUTO will immediately turn itself off as a safeguard.
- AUTO may be turned off by pressing RETURN after an AUTO number has been generated.

N.B.

During a DUMP, string and array variables will be displayed as the variable name followed by the dimensioned value of the variable. A '1' will be shown if the string or array has not yet been dimensioned.

OPTION 'B' FACILITIES AND COMMANDS - continued**#REPORT**

Atari BASIC allows looping to line numbers identified by variables (example: LINE=1000;GOTO LINE) which gives problems when renumbering. REPORT lists every loop to a variable. This lets you make a note and adjust such loops after renumbering. (Hint: if REPORT discovers a number of them, examine the start of the program - the variables are usually set up in the first few lines).

REPORT also highlights any references to non-existent line numbers.

#RENUMBER

Renumbers the program in memory. All GOTO, GOSUB, ON...GOTO, ON...GOSUB, LIST, RESTORE, TRAP and THEN references are also appropriately renumbered. The first new line number will be 100 and the increments will be 10. References to non-existent lines are renumbered to 32767 for ease of checking.

#RENUMBER s,i Renumbers the program in memory. Values should be entered for 's' and 'i'. The first renumbered line will be 's' and the increments will be 'i'.

#RENUMBER s

Renumbers the program in memory. The first renumbered line will be 's' and the increments will be 10.

#RENUMBER ,i

Renumbers the program in memory. The first renumbered line will be 100 and the increments will be 'i'.

#RENUMBER.F

The addition of the letter 'F' to any of the RENUMBER options indicates lines which can cause the problems described in REPORT in this section.

N.B.

If the TRAP command is used during a BASIC program, a line such as TRAP 40000 may be used to disable the trap command. After the program has ended, the line will be filled with the message: ERROR report during #REPORT or #RENUMBER and will be renumbered to 32767. This is still a legitimate line number and will still work correctly after a RENUMBER.

OPTION 'C' FACILITIES AND COMMANDS

When Option 'C' is chosen and loaded, all the facilities just described for Options 'A' and 'B' are available, together with these additional facilities:

- @BSAVE x,y** Saves an object file currently in memory between locations 'x' and 'y'. Values should be entered for 'x' and 'y'. These may be either hexadecimal, in which case both values should be preceded by \$, or decimal.
- @BLOAD** Loads an object file from cassette into memory. The low and high memory addresses will be shown.
- @BVERIFY** Examines a binary file saved on cassette and verifies that it will reload reliably.
- @VAL x** Provides instantaneous hexadecimal, decimal and binary number conversions. 'x' is the number for which conversions are required. Hexadecimal numbers should be preceded by \$; binary numbers by %; decimal numbers may be entered as they are.
- @POINTER x** Shows the pointer value of two memory bytes when 'x' is the memory location for which the pointer value is required.

ERROR MESSAGES

You may make a mistake while using ACE so you'll be pleased to hear ACE tells you what it is in plain English! Here are the ACE error messages.

Syntax ERROR!
You've entered a command which ACE doesn't understand. You've asked for a facility which isn't loaded. Check the command against these instructions or reload ACE for the required facility.

VALUE ERROR!
You've entered a command which contains an illegal value or you've asked for a RENUMBER which will go higher than 32766. Check your command against these instructions or choose a lower starting line number and/or a smaller increment.

CASSETTE I/O ERROR!
ACE can't load a program due to an error, probably on the tape. Make sure the tape was rewound and your tape head is clean, then try loading again.

VERIFY ERROR!
ACE has found an error on the SAVE'd program on tape.

INSUFFICIENT MEMORY!
If this occurs while loading a program, the program is too large for the available memory. If it occurs while loading a program, there isn't enough memory to alter the current program or there are more than 126 variables in the program. Try to reduce the number of variables.

NOT BINARY FILE!
You've used BLOAD for a non-binary program. Check you have the right cassette and you're loading from the right tape position.